

# The Automatic Generation of Piano Accompaniment from Four-Part Hymns

Ha Kyung Kong

The goal of this research project was to automate the generation of a piano accompaniment based on four-part hymns. The autogeneration is done by converting the four-part hymns and lead sheets that are written in .mxml (Music XML) format into a fully-written-out piano sheet music in .pdf format and .midi format.

“Our local church family is relatively small (<50 members). Although we have a few folk who play the piano and do provide special music from time-to-time, we do not currently have anyone who is comfortable accompanying our congregational singing via piano on an ongoing basis. This has been my biggest challenge over the past 2-1/2 years as music leader.”

Ken Rigby<sup>1</sup>

congregations have to struggle to start a hymn without any accompaniment. Some of these churches have people who play the piano. But playing a hymn is very different from playing a classical piece, and the pianist might feel uncomfortable reading an unfamiliar format of music in front of the congregation.

## 1. Introduction

Ken is a worship leader at a small church and shares a problem that many worship leaders face: the lack of an accompanist for congregational singing. Some worship leaders download mp3 and midi files to fill the musical gap and some churches might even buy a m.a.c. (musical accompaniment for church) that has hundreds of pre-recorded hymns. However, buying mp3 files or a m.a.c. might be too costly for small churches, and some

The Piano Accompaniment Generator hopefully addresses this problem by providing a style of hymn accompaniment that is familiar to classically trained pianists. By writing out the exact piano accompaniment, it lightens the burdens of the accompanists and broadens the range of prospective accompanists. The music can also be transposed for the comfort of the accompanist and the congregation. Hymns are often in keys that have more sharps or flats compared to classical music, so the transposition might help the accompanists play the hymns more easily.

---

<sup>1</sup> Rigby, Ken. "Accompaniment Alternatives When a Pianist/organist Is Not Available." *The Accidental Hymnologist*. 5 Feb. 2013. Web. 10 May 2013.

## 2. Preparation

A survey was given out to professors in the music department, accompanists in churches, student accompanists, and keyboardists in worship bands at Calvin College. The survey included questions on the accompaniment styles with seven different styles given as examples. Different styles included block chords, arpeggios, and syncopated rhythms (See Figure 1.a).

Then different techniques and advices for the beginning accompanists were asked, followed by a question about the improvisational chord progressions that the accompanists use to make the accompaniment smoother (See Figure 1.b).

After the surveys had been recollected, a general pattern and tips for accompanying were extracted from the surveys. The compiled answers to the surveys were then used for guiding the internal representation of the music and the styles of accompaniment. I had initially planned to use the chord progressions that were given to enhance the naturalness of the accompaniment, but could not work on the chord progressions due to time limitations.

Accompanist Survey

1. Accompaniment styles

If you know any additional styles, please write it on the sheet music.

Figure 1.a

Here is the general summary of the compiled answers to the survey.

1. Which techniques do you use most frequently as an accompanist? (ex. Fill ins, Patterns -> Ways/places to add 2nds, add 7ths on the dominant chord)

### ADDED NOTES

- Add2 (8ths, to tonic and IV chords)
- Added 4<sup>th</sup> without the third
- Add 6ths in major triads, min 7ths in minor triads (in dominant position in the right hand)

### RESOLUTIONS

- Secondary dominant – V, V7 , V65, vii dim resolve to V6
- 4-3 and 2-1 suspensions at the end of phrases
- Step down or step up as leading chord

### PATTERNS

- Broken chords or arpeggio patterns
- Supply rhythm, reinforce the harmony or enrich with added forms

### ALTERNATIVE CHORDS

- Alternative chords (ex. In key of C major, F ↔ Dm, C ↔ Am)

### OTHERS

- Tonic expansion -> expanding even move.

#### 4. Chord progressions. (mark only-bass notes by [ ])

Progressions	Optional chords (ex. C-G/B)	Progressions	Optional chords (ex. A/C#-[Bb]-Bm)
I - V		V - vi	
I - IV		vi - IV	
I - V		I (long tonic)	
I - vi		V (long domin.)	
ii - V			
III - vi			
IV - V			
V - I			
V - vi			

Progressions	Optional chords	Progressions	Optional chords
i - iv			
i - VI			
ii <sup>o</sup> - V			
III - VII			
IV -			
V - I			
VI - III			

Figure 1.b

2. Do you have any advices for beginning accompanists? (ex. leaving certain notes/chords out -> you can leave a tone out of the left hand if it's already in the melody)

### GENERAL

- Simple is the best
- A rest is as important as a note, keep space throughout
- Use dissonance wisely (ex. Don't use chords that are dissonant to the melody)
- Listen for clear pedaling – change pedals every time the harmony changes, and sometimes even when it doesn't
- Lighten the texture or voicing of chords the lower you go on the keyboard
- Always go for a steady beat before trying to embellish the accompaniment

### PROGRESSIONS

- Don't repeat/overuse unique chord progressions
- Think how a chord leads to another/ progression
- Use 2nds of the bass note to make diminished chords smooth

### VOICING

- Keep common tones on top
- Leave out the 3<sup>rd</sup> or 5<sup>th</sup> in Dominant 7<sup>th</sup> chords
- Avoid using the 5<sup>th</sup> of the chord except on weak beats.
- Root is best, 3<sup>rd</sup> may be used sparingly if not doubling the melody note.
- Do not double the leading tone

After giving out the surveys to the accompanists, I started to collect musicxml files from [www.hymnary.org](http://www.hymnary.org). Most of the musicxml files that were used for initial analysis were autogenerated from Noteworthy Composer. A typical musicxml file starts with a header section containing the general information, such as the title and the composer, of the hymn. It also contains a section for the appearance of the music as

well as the general information about the parts(SA and TB for hymns), such as the instrumentation, time signature, and the key signature. After defining the different parts, notes and rests are added to each part.

### 3. System design

#### I. Parsing the attributes from the XML file

Piano Accompaniment Generator (PAG) creates a tree of all the elements in the musicxml file using the ElementTree module in Python. It looks through the header element for general information, such as the title and the composer of the piece. Then it goes through the list and finds the note lengths, time signature, key signature, etc. from the attributes element. These features are used later to store the chords in an internal representation of the music.

#### II. Parsing the notes from the XML file/ Creating a chord list from the hymn

The PAG goes through the notes and creates a note element from each note with four attributes - pitch, octave, duration/type, alter. A list of four notes played or sung at the same time represents a chord (SATB) and is analyzed later for generating the accompaniment.

#### PITCH

The range of pitches is (C, D, E, F, G, A, B). I indicated rests by the pitch 'R' and non-existent or uninitiated notes by the pitch 'H.'

#### OCTAVE

The octaves in hymns generally range from 2 to 5. C4 notes the middle-C on the piano. Octave is explicit in musicxml but implicit in Lilypond; when Lilypond is given a set of notes [c, g], it chooses the note 'g' closest to the 'c.'

Choosing the octave for the accompaniment was harder than expected. I usually converted the pitches into ordinals to see the relationship between the notes. By comparing two input notes, I could decide whether a note belonged in the soprano line or the alto line based on which note was higher. However, the English alphabet starts with an 'a,' while musical notes start with a 'c.' If an octave of notes were listed in order it would be C4, D4, E4, G4, A4, B4, C5. Note that A and B are in the octave 4 not 5. Thus, I had to make exceptions for these cases by wrapping-around the octaves.

*DURATION / TYPE*

Each element in the list represents the shortest note in the piece. In other words, the length of a chord list for a measure with time signature 3/4 would be 3 if the shortest note is a quarter note, 6 if an eighth note, 12 if a sixteenth note, etc. Typically, the shortest note in a hymn is an eighth note, but some hymns have sixteenth notes. Because the first bar may be a pick up bar (an incomplete bar that is completed by the last measure), I had to extract the duration of a measure by going the second measure and adding the duration for all the notes in the measure. Since every element has the length equal to the shortest length in the piece, the duration of each note is marked by the type of the note: 'hold' or 'end.' For example, in a piece with the shortest note of an eighth note, a dotted quarter note would be represented by two 'hold's and an 'end.'

*ALTER*

Alter element indicates the accidentals of the note. The alter for a note with a flat is -1, a sharp 1, and a neutral is 0. I created a dictionary to facilitate the conversion between the symbols (b, #, -) and the numbers (-1, 1, 0). The accidentals are especially important in analyzing the quality of the chords, and they are also indicated in the pitch value of the note. The seemingly

redundant element of alter is useful in the creation of the output musicxml file.

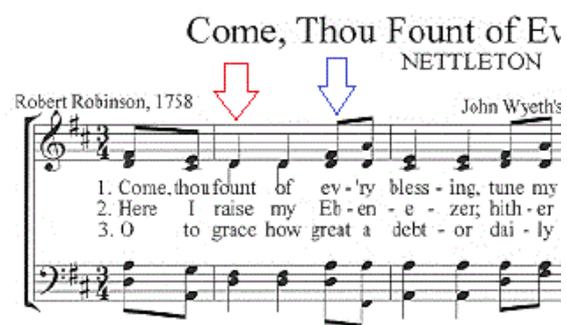
For example, notes in the soprano line in Figure 2 would look as follows:

*The first note, pointed to by a red arrow*

Pitch: D  
 Octave: 4  
 Type: held  
 Alter: 0

*The second note, pointed to by a blue arrow*

Pitch: F#  
 Octave: 4  
 Type: end  
 Alter: 1



**Figure 2**

III. Creating a chord list for the accompaniment

The third step involves analyzing the notes that are in a chord and adding an appropriate accompaniment depending on the style that has been selected. The PAG first passes down the four notes in a chord to a findChord function, which returns a tuple that consists of the chord pitch and the chord quality- ex. ("C", "major"). Then it iterates through the list of chords to see how long a chord lasts. The chord tuple and the duration are passed on to another function that creates an accompaniment chord list. The user can select the style of the accompaniment from a list of options. The accompaniment consists of two subparts: one for the right hand and one for the left

hand. The number of lists in each hand depends on the style of accompaniment. A simple arpeggio accompaniment only requires one list for the left hand, while a block chord accompaniment requires three.

#### IV. Outputting to different files

A new musicxml is produced from the accompaniment chord list. Then musicxml2ly script transforms the musicxml file into a Lilypond file. Changes are made to the Lilypond file including transposition and the setting of tempo before it is used to produce a midi file and a PDF file.

### 4. User Interface

I chose to format my project as a web application to make it accessible to worship leaders in small churches and anyone else who might want to use it. The server had to be set up so that it could handle Python script since the PAG was written in Python. The website is divided into three parts.

#### *PART ONE*

At first, the user enters a hymn tune name in the HTML form. The CGI program is then called to extract the values passed by the HTML form to a python script. The script reads a musicxml.csv file that has a table of all the tune names, first line of the hymn, and the URL of the musicxml file. When the script finds a matching tune name, it returns the hymn URL.

#### *PART TWO*

The user fills out the forms regarding the style, tempo, and the key of the accompaniment. The CGI program extracts this information and sends the newly gained information and the previously computed hymn URL to a Python script that generates the appropriate musicxml file.

#### *PART THREE*

The newly created files of different formats are stored in the tmp directory. The links for downloading the musicxml, midi file, and the PDF files are provided on the last result page with the embedded PDF file displayed at the bottom of the page.

### 5. Testing

Although I started out with musicxml file that had been generated from Noteworthy Composer, almost all of the musicxml files from the musicxml.csv file had been generated from another music software, Finale. So I had to test if the program was functional for files that had been generated from Finale as well as other musical software programs.

One major difference between the Final generated files and Noteworthy Composer generated files was the notation of the notes in a measure.

Noteworthy Composer generated files list soprano notes and add the alto notes by using chord elements.

Example:

```
<note>
  <pitch>
    <step>C</step>
    <octave>4</octave>
  </pitch>
  <duration>1</duration>
</note>
```

```
<note>
  <chord/>
  <pitch>
    <step>E</step>
    <alter>-1</alter>
    <octave>4</octave>
  </pitch>
  <duration>1</duration>
</note>
```

On the other hand, Finale generated files list one soprano note, use a backup element to

go back before the soprano note, then list an alto note.

Example:

```
<note color="#000000" default-x="15">
  <pitch>
    <step>D</step>
    <octave>4</octave>
  </pitch>
  <duration>512</duration>
  <instrument id="P1-I1" />
  <voice>1</voice>
  <type>half</type>
  <stem>up</stem>
  <staff>1</staff>
</note>

<backup>
  <duration>512</duration>
</backup>

<note color="#000000" default-x="15">
  <pitch>
    <step>B</step>
    <octave>3</octave>
  </pitch>
  <duration>512</duration>
  <instrument id="P1-I1" />
  <voice>2</voice>
  <type>half</type>
  <stem>up</stem>
  <staff> 1</staff>
</note>
```

Sometimes the files list all the soprano notes in the measure, then use a backup element to go to the beginning of the measure, then list all the alto notes in the measure.

So overall, the PAG has to be able to handle the backup element liberally, and deal with the chord elements when the files used them.

I tested the program by going through the list of hymns in the musicxml.csv file. When a hymn was successfully converted into a musicxml file, I viewed the file in Musescore to check for any flaws in the file. After the Musescore scores had been approved, a Lilypond file was generated from the musicxml files for further testing.

## 6. Results

The Piano Accompaniment Generator takes in musicxml files with desired style,

tempo, and key settings and generates output files in .ly, .pdf, and .midi format.

Currently, the program is able to convert some musicxml files, but it is not generalized to take in all possible musicxml files. One problem with parsing musicxml is that every music software program (or even the same program) uses a different formatting for musicxml.

The conversion between musicxml and Lilypond also caused various problems. The script musicxml2ly is still in development and is very selective on what kinds of musicxml it allows to be converted into Lilypond files. Thus, I had to format the output musicxml files in a specific way so that the Lilypond script could handle them.

So overall, the PAG works the most effectively on clean Sibelius generated files. However, the musicxml files usually diverge from the standards, making it almost impractical to have a program that only works on clean files.

However, I think the outcome of the program is satisfying since it showed the possibility of the conversion between a four-part hymn and a fully written out piano, which could help small churches all over the world.

## 7. Future perspectives

Future work on the project includes the generalization of the parser to address different formats of musicxml files and more sophisticated functions on choosing the chord progression of the hymn.

Since I have collected different improvisational accompaniments, the project could be enhanced to have a function that adds transitional chords in places where the original chord progression is too plain. Besides filling in the gaps between the chords, transitional chords could also change the atmosphere or the genre of the music by adding jazz chords to the accompaniment.